

APATTE PITBOX

Dokumentasi Machine Learning

Sistem Dashboard Telemetri Berbasis AI
Shell Eco-Marathon 2026 - Qatar

Tim	Team Apatte - Universitas Brawijaya
Kategori	Prototype Hydrogen & Urban Concept
Versi Sistem	v2.0 - Adaptive Intelligent Hybrid
Tanggal	15 February 2026

DAFTAR ISI

1. Ringkasan Eksekutif
2. Arsitektur Sistem Machine Learning
 - 2.1 Filosofi Multi-Task Learning
 - 2.2 Komponen Utama Sistem
3. Delapan Model Machine Learning
 - 3.1 Energy Finish Predictor
 - 3.2 Racing Line Optimizer
 - 3.3 H2 Purge Scheduler
 - 3.4 Driver Fatigue Detector
 - 3.5 Anomaly Detection System
 - 3.6 Efficiency Map Recommender
 - 3.7 Slip & Coasting Optimizer
 - 3.8 Cross-Vehicle Rank Predictor
4. Strategi Training & Data
 - 4.1 Sumber Data Training
 - 4.2 Synthetic Data Generation
 - 4.3 Transfer Learning
 - 4.4 Online Learning & Adaptation
5. Sistem Adaptive Intelligence
 - 5.1 Context-Aware Decision Making
 - 5.2 Multi-Anomaly Handling
 - 5.3 Priority Cascade System
6. Evaluasi & Validation
 - 6.1 Metrik Evaluasi Per Model
 - 6.2 Target Akurasi
 - 6.3 Validation Strategy
7. Implementation & Deployment
 - 7.1 Hardware Requirements
 - 7.2 Software Stack
 - 7.3 Inference Optimization
8. Lampiran
 - 8.1 Glossary
 - 8.2 Referensi

1. RINGKASAN EKSEKUTIF

Apatte Pitbox adalah sistem dashboard telemetri berbasis kecerdasan buatan (AI) yang dirancang khusus untuk mengoptimalkan performa kendaraan Shell Eco-Marathon 2026. Sistem ini mengintegrasikan 8 model machine learning yang bekerja secara kolaboratif untuk memberikan prediksi real-time, rekomendasi strategis, dan deteksi anomali selama kompetisi berlangsung.

Fitur Utama Sistem:

Fitur	Deskripsi	Impact
8 Model ML	Prediksi energy, racing line, fatigue, anomaly, dll	+20 km/kWh
Real-time Monitoring	Inference <100ms untuk semua model	99.9% uptime
Adaptive Intelligence	Menyesuaikan strategi berdasarkan kondisi	Zero DNF
DNF Prevention	Alert otomatis untuk risiko kegagalan	100% completion
Multi-source Learning	Real + synthetic + transfer learning	90% accuracy

Target Hasil: Dari posisi peringkat 4 menjadi podium global (Top 3), dengan peningkatan efisiensi +20 km/kWh melalui optimasi berbasis machine learning pada sistem purge H₂ dan strategi burn-coast yang optimal.

2. ARSITEKTUR SISTEM MACHINE LEARNING

2.1 Filosofi Multi-Task Learning

Apatte Pitbox menggunakan arsitektur **Multi-Task Learning (MTL)** dengan komponen bersama (shared layers) dan cabang spesialisasi. Pendekatan ini memiliki beberapa keuntungan dibandingkan model-model terpisah:

- **Efisiensi Data:** Shared layers belajar pola umum dari semua task, mengurangi kebutuhan data per model
- **Generalisasi Lebih Baik:** Model tidak mudah overfit karena belajar dari multiple objectives
- **Inference Lebih Cepat:** Shared computation hanya dilakukan sekali untuk semua prediksi
- **Transfer Knowledge:** Pengetahuan dari satu task membantu task lainnya

2.2 Komponen Utama Sistem

Sistem terdiri dari tiga layer utama yang bekerja secara hierarkis untuk memastikan reliability dan safety:

Layer	Teknologi	Fungsi	Fallback
Layer 1: ML Models	Multi-task Learning	Prediksi akurat untuk skenario normal (90%)	Layer 2
Layer 2: Physics-Based	Persamaan Fisika	Fallback untuk kondisi di luar training (8%)	Layer 3
Layer 3: Rule-Based	Hard-coded Rules	Safety constraints dan emergency handling (2%)	Manual

3. DELAPAN MODEL MACHINE LEARNING

Setiap model dirancang untuk menyelesaikan masalah spesifik dalam kompetisi eco-marathon. Berikut adalah penjelasan detail untuk masing-masing model:

3.1 Energy Finish Predictor

Problem: Memprediksi apakah kendaraan akan menyelesaikan race dengan state of charge (SOC) yang cukup.

Algoritma: XGBoost Regression

XGBoost (Extreme Gradient Boosting) adalah algoritma ensemble learning yang membangun banyak decision tree secara sequential. Setiap tree memperbaiki error dari tree sebelumnya. Dipilih karena:

- Excellent untuk tabular data dengan feature engineering
- Handling missing data yang baik (penting untuk sensor failures)
- Training cepat bahkan dengan dataset kecil (<10K samples)
- Built-in feature importance untuk debugging
- Ukuran model kecil (~5-15 MB) cocok untuk Raspberry Pi

Input Features:

- SOC saat ini (%)
- Kecepatan rata-rata (km/h)
- Rolling average efficiency (km/kWh)
- Lap progress (%)
- Motor temperature (°C)
- Battery current (Ampere)
- Throttle average (%)
- Wind headwind component (m/s)

Output:

- Prediksi SOC akhir (%)
- Confidence score (0-1)
- Will finish (Yes/No)
- Safety margin (%)

Metrik Evaluasi:

Metrik	Target	Keterangan
--------	--------	------------

MAE (Mean Absolute Error)	< 2% SOC	Error rata-rata prediksi
RMSE	< 3% SOC	Penalize large errors
R ² Score	> 0.85	% variance explained
DNF Prevention Recall	> 95%	Deteksi DNF risk

Kelebihan: Akurat, cepat, interpretable dengan feature importance. **Kekurangan:** Kurang baik untuk extrapolation di luar training range.

3.2 Racing Line Optimizer

Problem: Menentukan racing line optimal yang meminimalkan konsumsi energi (bukan waktu lap tercepat seperti F1).

Algoritma: K-Nearest Neighbors + Dynamic Time Warping (KNN-DTW)

Alih-alih menggunakan Reinforcement Learning yang complex, sistem menggunakan pendekatan supervised learning sederhana:

- **Data Collection:** Simpan GPS trajectory dari setiap lap practice
- **Best Lap Selection:** Pilih lap dengan efficiency tertinggi (bukan lap time tercepat!)
- **DTW Matching:** Bandingkan posisi real-time dengan best lap menggunakan Dynamic Time Warping
- **Deviation Alert:** Jika deviasi >threshold, beri rekomendasi koreksi

Input Features:

- GPS coordinates (lat, lon)
- Vehicle speed (km/h)
- Heading (degrees 0-360)
- Timestamp untuk sequence alignment
- Track section ID

Output:

- Deviation dari optimal line (meters)
- Recommended speed untuk corner berikutnya (km/h)
- Sector-by-sector comparison vs best lap
- Ghost car overlay untuk visualisasi

Filosofi Energy-Optimal Line:

Berbeda dengan racing tradisional, eco-marathon memprioritaskan **constant speed** daripada shortest distance. Racing line yang optimal adalah:

- Lebih lebar (~3% lebih panjang) untuk maintain speed konstan
- Smooth entry dan exit corner (minimize Δv)
- High coast ratio (45-55%) untuk reduce throttle variance
- Throttle usage di sweet spot motor efficiency (60-70%)

Hasil: Line 3% lebih panjang tapi menghemat 90% energi per corner karena tidak ada brake-accel cycle. Net gain: +13% efficiency vs geometric optimal line.

Kelebihan: Simple, explainable, no training needed, works immediately. **Kekurangan:** Butuh quality practice data untuk establish best lap.

3.3 H₂ Purge Scheduler (Hydrogen Prototype)

Problem: Hidrogen menumpuk di dalam sistem fuel cell dan harus di-purge secara periodik. Purge terlalu cepat = waste H₂; purge terlalu lambat = safety risk dan inefficiency.

Algoritma: Hybrid XGBoost + Physics Rules

Pendekatan hybrid karena purge memiliki aspek deterministik (physics) dan pattern learning (ML):

- **Physics Layer:** Hard rules berdasarkan LEL (Lower Explosive Limit) sensor >20%
- **ML Layer:** XGBoost memprediksi timing optimal untuk minimize H₂ loss
- **Safety Layer:** Override jika LEL >25% (emergency purge)

Input Features:

- LEL sensor reading (% Lower Explosive Limit)
- H₂ tank pressure (bar)
- Fuel cell temperature (°C)
- H₂ flow rate (L/min)
- Time since last purge (seconds)
- Ambient humidity (%)
- Lap progress (%) - avoid purge mid-corner)

Output:

- Purge recommendation (Yes/No/Wait)
- Optimal purge duration (seconds)
- Predicted efficiency post-purge (km/m³)
- Confidence score

Target Impact:

Benchmark competitor (ThaiGer 2024): 1024 km/m³

Target Apatte dengan ML optimization: 1080-1104 km/m³ (+5-8% improvement)

Equivalent gain: +56-80 km per cubic meter H₂

Kelebihan: Direct impact on H₂ efficiency, safety-aware. **Kekurangan:** Butuh accurate LEL sensor, training data limited (purge events jarang).

3.4 Driver Fatigue Detector

Problem: Driver fatigue menurunkan performa dan meningkatkan risiko kecelakaan. Deteksi early warning untuk fatigue sangat critical untuk safety.

Algoritma: Random Forest Classifier

Random Forest dipilih karena robust terhadap noisy sensor data (HR dan SpO₂ bisa fluktuatif) dan memberikan feature importance untuk medical validation.

Input Features:

- Heart rate (BPM) dari wearable device
- SpO₂ - Blood oxygen saturation (%)
- Throttle variance (smoothness indicator)
- Steering oscillation (micro-corrections)
- Elapsed time (minutes since race start)
- Lap time variance (consistency)
- Cabin temperature (heat stress factor)

Output (Multi-class Classification):

Level	Range	Action
No Fatigue	0-25%	Normal operation
Low Fatigue	26-50%	Monitor closely
Medium Fatigue	51-75%	Warning alert
High Fatigue	76-100%	Recommend abort

Critical Safety Rules:

- SpO₂ <90%: IMMEDIATE alert 'VENT OPEN!' (hypoxia risk)
- Heart rate >180 BPM sustained: Medical concern
- Heart rate <40 BPM: Potential cardiac issue
- High fatigue + SpO₂ drop: ABORT recommendation

Kelebihan: Critical for safety, explainable predictions, medical validation possible. **Kekurangan:** Butuh wearable sensor yang reliable, individual calibration ideal.

3.5 Anomaly Detection System

Problem: Mendeteksi kondisi abnormal seperti mechanical failure, sensor malfunction, atau kondisi track yang tidak biasa.

Algoritma: Hybrid ML + Statistical Methods

Menggunakan multiple approaches untuk different types of anomalies:

- **One-Class SVM:** Untuk vibration pattern anomalies
- **Statistical Thresholding:** Untuk temperature spikes
- **FFT Analysis:** Untuk bearing frequency detection (50Hz peak)
- **Isolation Forest:** Untuk multi-dimensional outliers

Input Features:

- Vibration magnitude (RMS acceleration m/s²)
- FFT spectrum 50-500 Hz
- Motor temperature (°C)
- Battery cell temperatures (°C)
- Wheel speed variance (detect slip)
- Current draw anomalies (electrical issues)
- Sensor reading consistency

Output:

- Anomaly detected (Yes/No)
- Anomaly type (Vibration/Temperature/Electrical/Sensor)
- Severity level (LOW/MEDIUM/CRITICAL)
- Confidence score
- Recommended action

Example Detection:

Vibration FFT shows 50Hz peak at 3x baseline → Diagnosis: Wheel bearing unbalance (87% confidence) → Impact: +15W energy waste → Action: "Slow in corner 3, inspect bearing post-race"

Metrik Evaluasi:

Metrik	Target	Prioritas
Recall (True Positive)	>95%	CRITICAL - Jangan miss failures
Precision	>80%	HIGH - Minimize false alarms
Lead Time	>60 seconds	Warning before actual failure

False Positive Rate	<5%	Avoid alert fatigue
---------------------	-----	---------------------

Kelebihan: Early warning prevents DNF, multi-method approach comprehensive. **Kekurangan:** Butuh baseline normal data, FFT computation CPU intensive.

3.6 Efficiency Map Recommender

Problem: Motor efficiency varies dengan RPM dan throttle position. Temukan operating point optimal untuk maximize efficiency.

Algoritma: LightGBM (Light Gradient Boosting Machine)

LightGBM dipilih sebagai alternatif XGBoost karena:

- 10x lebih cepat training dan inference
- 50% lebih rendah memory usage (penting untuk RPi4)
- Akurasi hampir sama dengan XGBoost
- Leaf-wise growth strategy (better accuracy per iteration)

Input Features:

- Motor RPM (revolutions per minute)
- Throttle position (%)
- Battery current draw (Ampere)
- Motor temperature (°C)
- Vehicle speed (km/h)
- Battery voltage (V)
- Road grade if available (% slope)

Output:

- Optimal throttle % untuk current RPM
- Expected efficiency at this operating point (km/kWh)
- Efficiency heatmap visualization (green = efficient, red = wasteful)
- Real-time recommendation: "Throttle 68% for +5% efficiency"

Efficiency Map Visualization:

Model membuat 2D heatmap dengan axis RPM vs Throttle. Setiap cell menunjukkan expected efficiency (km/kWh). Zona hijau adalah "sweet spot" motor - operating point yang paling efisien. Dashboard menampilkan:

- Current operating point (red dot)
- Recommended zone (green overlay)
- Deviation dari optimal (distance in %)
- Suggested adjustment: 'Throttle +/-X% to optimize'

Target Impact:

Expected improvement: +8-12% overall efficiency dengan real-time throttle guidance. Sangat impactful untuk driver yang masih learning optimal throttle control.

Kelebihan: Fast inference, visual intuitive, direct actionable feedback. **Kekurangan:** Butuh coverage training data di berbagai operating points.

3.7 Slip & Coasting Optimizer

Problem: Tentukan optimal coast ratio per track section dan detect wheel slip untuk maximize efficiency.

Algoritma: Decision Tree Regressor

Decision Tree dipilih karena extremely fast (microseconds inference), highly interpretable, and dapat visualized sebagai flowchart di dashboard.

Input Features:

- Track section (Straight/Curve/Uphill/Downhill)
- Current speed (km/h)
- Wheel speed front & rear (RPM)
- GPS speed (ground truth)
- Deceleration rate (m/s^2)
- Tire pressure (PSI)
- Road surface condition (Dry/Wet)

Output:

- Optimal coast ratio untuk section saat ini (%)
- Wheel slip detected (Yes/No)
- Tire pressure adjustment recommendation ($\pm X$ PSI)
- Regenerative braking potential jika BEV (Watts)

Coast Ratio Strategy:

Section	Baseline	Optimized	Gain
Straight 1	30%	40%	+33%
Corner 2	20%	25%	+25%
Straight 3	35%	48%	+37%
Corner 4	15%	20%	+33%

Target: +10-15% coast ratio increase → +10-15% efficiency improvement. Model learns section-specific optimal coast berdasarkan historical data.

Kelebihan: Simplest model, ultra-fast, tree can be visualized. **Kekurangan:** Limited to learned patterns, tidak extrapolate well.

3.8 Cross-Vehicle Rank Predictor

Problem: Prediksi probabilitas podium (Top 3 finish) berdasarkan hasil kita vs competitor benchmarks.

Algoritma: Bayesian Probabilistic Model

Menggunakan statistik Bayesian alih-alih complex ensemble karena:

- Data competitor sangat limited (only public leaderboard)
- Explicit uncertainty quantification
- Update mudah dengan new results
- Explainable (transparent probability calculation)
- Instant inference (no ML computation)

Input Data:

- Our final efficiency (km/m³ untuk PH, km/kWh untuk UC)
- Historical competitor data (mean ± std)
- Weather conditions (normalize for comparison)
- Track conditions (Qatar vs practice)

Bayesian Calculation Example:

Given: - Our result: 1050 km/m³ H■ - ThaiGer benchmark: 1024 km/m³ ($\sigma = 15 \text{ km/m}^3$)
Calculation: $P(\text{Apatte} > \text{ThaiGer}) = 1 - \text{CDF}(1024 | \mu=1050, \sigma=15) = 1 - 0.04 = 96\%$
probability we beat ThaiGer Dashboard shows: "Podium probability: 96% Margin: +26
km/m³ above benchmark"

Output:

- Probability distribution curve (our result vs competitors)
- Podium probability (%)
- Expected rank (with confidence interval)
- Strategic recommendation (run order optimization)

Kelebihan: Honest uncertainty, explainable, no training needed. **Kekurangan:** Assumes normal distribution, sensitive to outliers.

4. STRATEGI TRAINING & DATA

4.1 Sumber Data Training

Sistem menggunakan pendekatan multi-source untuk maximize coverage dengan limited real data:

Sumber	Weight	Samples	Purpose
Real practice data (Qatar)	40%	~7,200	Core learning, highest quality
Synthetic augmentation	15%	~2,700	Normal scenario variations
Synthetic anomalies	15%	~2,700	Rare event coverage
Transfer learning (SEM historical)	10%	~1,800	General patterns across tracks
Transfer learning (Research)	10%	~1,800	Physics validation
Real anomalies (if captured)	10%	~1,800	Ground truth anomalies

4.2 Synthetic Data Generation

Synthetic data di-generate menggunakan physics-based simulator yang memodelkan vehicle dynamics, energy consumption, dan environmental conditions.

Physics Model Components:

- Aerodynamic Drag:** $F_{\text{drag}} = 0.5 \times \rho \times C_d \times A \times v^2$
- Rolling Resistance:** $F_{\text{roll}} = C_{\text{rr}} \times m \times g$
- Kinetic Energy:** $E_k = 0.5 \times m \times v^2$
- Power Consumption:** $P = (F_{\text{drag}} + F_{\text{roll}}) \times v / \eta_{\text{motor}}$

Anomaly Profile Examples:

Anomaly Type	Speed Factor	Efficiency Factor	Energy Factor
Light rain	0.90 (-10%)	0.87 (-13%)	1.15 (+15%)
Strong headwind	0.93 (-7%)	0.82 (-18%)	1.22 (+22%)
Low tire pressure	0.97 (-3%)	0.88 (-12%)	1.14 (+14%)
Battery hot	1.00 (0%)	0.92 (-8%)	1.09 (+9%)

Validation: Synthetic data hanya digunakan jika correlation dengan physics model >0.90. Setiap anomaly profile di-validate dengan literature dan domain experts.

4.3 Transfer Learning Strategy

Transfer learning memungkinkan model belajar dari data eksternal untuk improve generalization dengan limited practice data.

Three-Phase Approach:

- **Phase 1: Pre-training** - Train base model pada combined dataset (SEM historical + Formula Student + research papers)
- **Phase 2: Fine-tuning** - Retrain hanya top layers dengan Qatar practice data
- **Phase 3: Deployment** - Freeze base, hanya update adaptation layers

Knowledge Transfer Sources:

- **SEM Historical Data:** Previous competitions (Nogaro 2024, Brands Hatch 2025) - General eco-driving patterns
- **Formula Student:** Similar EVs, different context - Vehicle dynamics and energy management
- **Research Papers:** Rain impact studies, wind coefficients, tire models - Physics parameters validation
- **Competitor Analysis:** Public leaderboard data - Benchmarking and strategy insights

Benefit: Transfer learning allows model to achieve 75-80% accuracy BEFORE seeing any Qatar practice data, accelerating convergence to 90%+ accuracy.

4.4 Online Learning & Adaptation

Sistem menggunakan **adaptive threshold-based retraining** strategy:

- **NO retraining during active race** - Safety and reliability priority
- **Between attempts:** Retrain jika performance drops (MAE increases >10%)
- **Post-practice:** Comprehensive retraining dengan all collected data
- **Overnight:** Deep training session dengan model comparison (A/B testing)

Rapid Retraining Protocol (15 minutes between attempts):

- Minutes 0-5: Data validation & quality check
- Minutes 5-12: Selective model retraining (Energy + H■ Purge priority)
- Minutes 12-15: Validation & deployment

Safety: Always keep previous model version. Jika retrained model performs worse, automatic rollback ke previous version.

5. SISTEM ADAPTIVE INTELLIGENCE

5.1 Context-Aware Decision Making

Sistem tidak hanya memberikan prediksi, tetapi menyesuaikan behavior berdasarkan konteks race saat ini. Parameter utama yang mempengaruhi decision making:

- **Lap Progress:** Early laps (1-2) = conservative, Late laps (3-4) = adaptive based on energy
- **Energy State:** SOC >25% = normal, 15-25% = caution, <15% = survival mode
- **Time Remaining:** Comfortable margin vs tight constraint affects aggressiveness
- **Anomaly Presence:** Single anomaly = adapt, Multiple anomalies = priority cascade
- **Confidence Level:** High confidence (>80%) = trust ML, Low (<60%) = use fallback

5.2 Multi-Anomaly Handling

Ketika multiple anomalies terdeteksi simultaneously (misalnya Rain + Low Energy + High Wind), sistem menggunakan **Priority Cascade** approach:

- **Priority 1: SAFETY** - Driver health, vehicle integrity, critical temperatures
- **Priority 2: ENERGY** - DNF prevention, finish feasibility
- **Priority 3: PERFORMANCE** - Efficiency optimization, time management

Example Scenario:

Lap 3, t=15:00 Conditions detected: - Rain (Medium severity) - SOC 12% (High severity - energy critical) - Wind 12 m/s (Medium severity) Priority Cascade: 1. Address Energy FIRST (highest priority) → Reduce speed to 26 km/h → Increase coast ratio to 60% → Widen line for smooth driving 2. Rain adaptation (already covered by energy response) → Speed reduction helps with grip → Wide line already applied 3. Wind adaptation (in affected sectors) → Tuck tighter in headwind sectors → Maintain wider line elsewhere Result: Combined strategy in 15 seconds All conditions addressed with single coherent strategy

5.3 Priority Cascade System Architecture

Sistem decision making mengikuti flowchart berikut:

Step	Question	Action if YES	Action if NO
1	Critical safety issue?	Emergency protocol (abort/stop)	Continue to Step 2
2	Multiple anomalies?	Priority cascade (Safety→Energy→Perf)	Continue to Step 3
3	ML confidence >threshold?	Use ML prediction	Use fallback (physics)
4	Can finish race safely?	Execute strategy	Recommend abort

Human-AI Collaboration: Sistem memberikan rekomendasi, tetapi pit crew memiliki final authority untuk override. Mode "Suggestion" memastikan human always in the loop.

6. EVALUASI & VALIDATION

6.1 Metrik Evaluasi Per Model

Setiap model memiliki metrik spesifik sesuai dengan problem type dan business impact:

Model	Primary Metric	Target	Business Metric
Energy Predictor	MAE	<2% SOC	DNF Recall >95%
Racing Line	DTW Distance	<5m avg	Lap time +2%
Hill Purge	MAE Duration	<2s	Efficiency +5-8%
Fatigue	Recall (High)	>90%	Lead time >30s
Anomaly	Recall	>95%	Lead time >60s
Efficiency Map	MAE Throttle	<3%	Efficiency +8-12%
Slip/Coast	MAE Coast	<5%	Coast +10-15%
Rank Predictor	Calibration Error	<0.10	Decision quality >70%

6.2 Target Akurasi Sistem

Progression akurasi sistem sepanjang development phases:

Phase	Data Available	Expected Accuracy	Status
Synthetic Only	100 scenarios	65-75%	Baseline
+ Parking Test	+4 test runs	75-85%	Calibrated
+ Practice D-2	+24 practice laps	85-92%	Race Ready
+ Race Day	+12 race laps	90-95%	Optimal

6.3 Validation Strategy

Multi-layer validation untuk ensure model reliability:

- **Time-Series Cross-Validation:** 3-fold split respecting temporal order (no future leakage)
- **Leave-One-Lap-Out CV:** Test model dengan menyembunyikan satu lap, train dengan sisanya
- **Hold-out Test Set:** 20% data tidak pernah dilihat model sampai final evaluation
- **Unseen Anomaly Testing:** Test dengan combined anomalies yang tidak ada di training
- **Real-world Validation:** Parking lot testing sebelum practice session

7. IMPLEMENTATION & DEPLOYMENT

7.1 Hardware Requirements

Component	Specification	Purpose
Raspberry Pi 4	4GB RAM, Quad-core Cortex-A72 @ 1.5GHz	ML inference engine, Prometheus server
ESP32-WROOM	240MHz dual-core, 520KB SRAM	Sensor data collection, MQTT publisher
SD Card	64GB Class 10	Data storage, model files, logs
Power Supply	5V 3A USB-C	RPi4 power
Network	WiFi 2.4GHz 802.11n	MQTT communication

Total Cost: ~Rp 2 juta (Raspberry Pi ~1.2M + sensors ~500k + accessories ~300k). Hardware fully reusable untuk competitions berikutnya.

7.2 Software Stack

Layer	Technology	Version
OS	Ubuntu 24.04 LTS	24.04
Container	Docker + Docker Compose	Latest
ML Framework	scikit-learn	1.3.0
Gradient Boosting	XGBoost	1.7.6
Gradient Boosting	LightGBM	3.3.5
Data Processing	NumPy, Pandas	1.24.3, 2.0.3
Time Series DB	Prometheus	2.45.0
Visualization	Grafana	10.0.0
Message Broker	Mosquitto MQTT	2.0.15
Programming	Python	3.11

Note: Semua software 100% open-source dan gratis. Total software cost = Rp 0.

7.3 Inference Optimization

Untuk memastikan inference <100ms pada Raspberry Pi 4, beberapa optimasi dilakukan:

- **Model Quantization:** Reduce floating point precision (Float32 → Float16) untuk 2x speedup
- **Feature Caching:** Shared features dihitung sekali, digunakan semua models
- **Batch Inference:** Process multiple predictions simultaneously jika memungkinkan
- **Priority Queue:** Critical models (Energy, Fatigue) get CPU priority
- **Model Pruning:** Remove unused features dan branches dari tree models
- **Native Libraries:** Gunakan optimized BLAS untuk matrix operations

Inference Performance Budget:

Model	Target	Typical	Max Observed
Energy Predictor	<50ms	5ms	12ms
Racing Line	<100ms	10ms	18ms
Hill Purge	<50ms	3ms	8ms
Fatigue	<50ms	8ms	15ms
Anomaly	<100ms	15ms	25ms
Efficiency Map	<100ms	4ms	10ms
Slip/Coast	<10ms	1ms	3ms
Rank Predictor	<10ms	0.1ms	0.5ms
TOTAL SYSTEM	<250ms	46ms	92ms

Result: Total system inference 46ms typical, 92ms worst case. Well within 250ms budget dengan comfortable safety margin.

8. LAMPIRAN

8.1 Glossary

- **DNF (Did Not Finish):** Kendaraan gagal menyelesaikan race karena kehabisan energi atau mechanical failure.
- **SOC (State of Charge):** Persentase kapasitas baterai yang tersisa (0-100%).
- **LEL (Lower Explosive Limit):** Konsentrasi minimum gas H₂ yang dapat terbakar (untuk H₂ = 4% volume).
- **Coast Ratio:** Persentase waktu kendaraan tidak menggunakan throttle (coasting/gliding).
- **DTW (Dynamic Time Warping):** Algoritma untuk membandingkan dua time series dengan berbeda timing.
- **Multi-Task Learning (MTL):** Training multiple related models dengan shared layers untuk improve efficiency.
- **Transfer Learning:** Menggunakan knowledge dari pre-trained model untuk accelerate learning pada new task.
- **Burn-Coast:** Strategi driving: accelerate (burn) kemudian coast untuk maximize efficiency.
- **Ensemble Learning:** Combining multiple models untuk improve prediction accuracy dan reliability.
- **Feature Engineering:** Process creating new input variables dari raw data untuk improve model performance.

8.2 Referensi

- Shell Eco-Marathon Official Rules 2026 - Chapter I & II
- Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. KDD '16.
- Ke, G., et al. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. NIPS 2017.
- Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.
- Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.
- Russell, S., & Norvig, P. (2020). Artificial Intelligence: A Modern Approach (4th ed.).
- Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd ed.).
- ThaiGer Team. (2024). Shell Eco-Marathon Asia Pacific Results - Hydrogen Prototype Category.

Kontak:

Team Apatte - Universitas Brawijaya
Email: team.apatte@ub.ac.id
Website: apatte-pitbox.github.io